

# Package: rgoogleslides (via r-universe)

September 13, 2024

**Type** Package

**Title** R Interface to Google Slides

**Version** 0.3.2

**Author** Hairizuan Bin Noorazman

**Maintainer** Hairizuan Noorazman <hairizuanbinnoorazman@gmail.com>

**Description** Previously, when one is working with in the Google Ecosystem (Using Google Drive etc), there is hardly any good workflow of getting the values calculated from R and getting that into Google Slides. The normal and easy way out would be to just copy your work over but when you have a number of analysis to present with a lot of changes between each environment, it just becomes quite cumbersome.

**License** MIT + file LICENSE

**LazyData** TRUE

**Imports** htr (>= 1.1.0), jsonlite, assertthat, R6

**Suggests** knitr, rmarkdown, testthat

**RoxygenNote** 7.0.2

**Repository** <https://hairizuanbinnoorazman.r-universe.dev>

**RemoteUrl** <https://github.com/hairizuanbinnoorazman/rgoogleslides>

**RemoteRef** HEAD

**RemoteSha** 2b32821962ab543c87f56fea009a8e3215226b53

## Contents

add_create_image_request . . . . .	2
add_create_shape_request . . . . .	3
add_create_slide_page_request . . . . .	4
add_create_table_request . . . . .	5
add_create_video_request . . . . .	6
add_delete_object_request . . . . .	7
add_delete_table_column_request . . . . .	8

add_delete_table_row_request . . . . .	8
add_delete_text_request . . . . .	9
add_insert_table_columns_request . . . . .	10
add_insert_table_rows_request . . . . .	10
add_insert_text_request . . . . .	11
add_replace_all_shapes_with_image_request . . . . .	12
add_replace_all_text_request . . . . .	13
add_update_slides_position_request . . . . .	14
aligned_page_element_property . . . . .	14
authorize . . . . .	15
check_validity . . . . .	16
commit_to_slides . . . . .	16
create_data_table . . . . .	17
create_slides . . . . .	17
dataframe_convert . . . . .	18
get_endpoint . . . . .	18
get_google_drive_urls . . . . .	19
get_slides_properties . . . . .	19
get_slide_page_properties . . . . .	19
is.google_slide_request . . . . .	20
is.page_element_property . . . . .	20
page_element_property . . . . .	21
rgoogleSlides . . . . .	22

## Index 23

---

add\_create\_image\_request  
*Add a create image request*

---

### Description

Add a create image request

### Usage

```
add_create_image_request(
    google_slides_request = NULL,
    page_element_property,
    url,
    object_id = NULL
)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
page_element_property	A list that contains a page element property. The page element is to be generated by the page_element_property function in this package.
url	A character vector container an image url that is to be used to add image to the slides
object_id	(Optional) A character vector to name the object created instead of leaving it to Google

---

 add\_create\_shape\_request

*Add a create shape request*


---

**Description**

This function builds up the request for creating shapes within Googleslides via the API.

**Usage**

```
add_create_shape_request(
  google_slides_request = NULL,
  shape_type,
  page_element_property,
  object_id = NULL
)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
shape_type	A character vector that contains the shape type for the new shape that is to be created
page_element_property	A list that contains a page element property. The page element is to be generated by the page_element_property function in this package. IT IS COMPULSORY TO ADD WIDTH AND HEIGHT AS PART OF THE page_element_property
object_id	(Optional) A character vector to name the object created instead of leaving it to Google

**Examples**

```
## Not run:
library(rgoogleSlides)
rgoogleSlides::authorize()

# Define the presentation slide id (Can be retrieved from the url of the slides)
slides_id <- "<slide-id>"

slide_page <- page_element_property("p", 200, 300)
request <- add_create_shape_request(shape_type = "RECTANGLE", page_element_property = slide_page)
commit_to_slides(slides_id, request)

slide_page <- aligned_page_element_property("p", image_height = 200, image_width = 300)
request2 <- add_create_shape_request(shape_type = "STAR_5", page_element_property = slide_page)
commit_to_slides(slides_id, request2)

## End(Not run)
```

---

```
add_create_slide_page_request
      Add a create slide request
```

---

**Description**

Add a create slide request

**Usage**

```
add_create_slide_page_request(
  google_slides_request = NULL,
  insertion_index = NULL,
  layout_id = NULL,
  predefined_layout = NULL,
  object_id = NULL
)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
insertion_index	(Optional) A numeric vector on where the slide is to be added. If this value is not provided, the new slide will be added at the end of the slides.
layout_id	(Optional) A character vector that provides guidance on which layout the new slide is to follow. Either layout id or predefined layout will be used to create the template for the new slide. If layout id was not provided, a blank predefined layout will be provided instead.

predefined\_layout

(Optional) A character vector that provides guidance on which layout the new slide is to follow. Some of the values that would be available is as follows:

- BLANK
- CAPTION\_ONLY
- TITLE
- TITLE\_AND\_BODY
- TITLE\_AND\_TWO\_COLUMNS
- TITLE\_ONLY
- SECTION\_HEADER
- SECTION\_TITLE\_AND\_DESCRIPTION
- ONE\_COLUMN\_TEXT
- MAIN\_POINT
- BIG\_NUMBER

The default value for this parameter is BLANK

object\_id

(Optional) A character vector that is to be used to give names to new slides created via the slides API

### Examples

```
## Not run:
library(rgoogleSlides)
rgoogleSlides::authorize()

# Define the presentation slide id (Can be retrieved from the url of the slides)
slides_id <- "<slide-id>"

requests <- add_create_slide_page_request()
commit_to_slides(slides_id, requests)

requests2 <- add_create_slide_page_request(predefined_layout = "TITLE")
commit_to_slides(slides_id, requests2)

## End(Not run)
```

---

add\_create\_table\_request

*Add a create table request*

---

### Description

Add a create table request

**Usage**

```
add_create_table_request(
  google_slides_request = NULL,
  page_element_property,
  rows,
  columns,
  object_id = NULL
)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
page_element_property	A list that contains a page element property. The page element is to be generated by the page_element_property function in this package.
rows	A numeric vector with the row index of the data to be filled in the table. Only for tables
columns	A numeric vector with the column index of the data to be filled in the table. Only for tables
object_id	(Optional) A character vector to name the object created instead of leaving it to Google

---

```
add_create_video_request
  Add a create video request
```

---

**Description**

Add a create video request

**Usage**

```
add_create_video_request(
  google_slides_request = NULL,
  id,
  page_element_property,
  object_id = NULL
)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
id	The video source's unique identifier for this video. e.g. For YouTube video <a href="https://www.youtube.com/watch?v=7U3axjORYZ0">https://www.youtube.com/watch?v=7U3axjORYZ0</a> , the ID is 7U3axjORYZ0.
page_element_property	A list that contains a page element property. The page element is to be generated by the <code>page_element_property</code> function in this package.
object_id	(Optional) A character vector to name the object created instead of leaving it to Google

**Examples**

```
## Not run:
library(rgoogleSlides)
rgoogleSlides::authorize()

# Define the presentation slide id (Can be retrieved from the url of the slides)
slides_id <- "<slide-id>"

# Define the youtube video id (Can be retrieved from the url of the youtube video)
youtube_id <- "<youtube-id>"

page_element <- aligned_page_element_property("p", align = "full")
request <- add_create_video_request(id = youtube_id, page_element_property = page_element)
commit_to_slides(slides_id, request)

## End(Not run)
```

---

```
add_delete_object_request
      Add a delete object request
```

---

**Description**

Add a delete object request

**Usage**

```
add_delete_object_request(google_slides_request = NULL, object_id = NULL)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
object_id	A character vector of object ids that is to be deleted from the slides

---

`add_delete_table_column_request`*Add a delete table column request*

---

**Description**

Deletes a column from a table.

**Usage**

```
add_delete_table_column_request(  
  google_slides_request = NULL,  
  table_object_id,  
  row_index,  
  column_index  
)
```

**Arguments**

<code>google_slides_request</code>	(Optional) A Google Slides Request object which is used to manage requests to the API
<code>table_object_id</code>	The table to insert rows into.
<code>row_index</code>	The 0-based row index.
<code>column_index</code>	The 0-based column index.

---

`add_delete_table_row_request`*Add a delete table row request*

---

**Description**

Deletes a row from a table.

**Usage**

```
add_delete_table_row_request(  
  google_slides_request = NULL,  
  table_object_id,  
  row_index,  
  column_index  
)
```



**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
table_object_id	The table to delete a row from.
row_index	The 0-based row index.
column_index	The 0-based column index.

---

 add\_delete\_text\_request

*Add a delete text request*


---

**Description**

Add a delete text request

**Usage**

```
add_delete_text_request(
  google_slides_request = NULL,
  object_id,
  row_index = NULL,
  column_index = NULL,
  start_index = NULL,
  end_index = NULL,
  type = "ALL"
)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
object_id	A string to remove text from. You can only remove text contained in tables and shapes
row_index	(Optional) A numeric vector of row to remove the text from. Only needed if you are deleting text from a table
column_index	(Optional) A numeric vector of column to remove the text from. Only needed if you are deleting text from a table
start_index	(Optional) The optional zero-based index of the beginning of the collection. Required for SPECIFIC_RANGE and FROM_START_INDEX ranges.
end_index	(Optional) The optional zero-based index of the end of the collection. Required for SPECIFIC_RANGE delete mode.
type	The type of range. Can be the following FIXED_RANGE, FROM_START_INDEX, ALL. The default value is ALL

---

`add_insert_table_columns_request`*Add an insert table column request*

---

**Description**

Add an insert table column request

**Usage**

```
add_insert_table_columns_request(  
  google_slides_request = NULL,  
  table_object_id,  
  row_index,  
  column_index,  
  insert_right = TRUE,  
  number  
)
```

**Arguments**

<code>google_slides_request</code>	(Optional) A Google Slides Request object which is used to manage requests to the API
<code>table_object_id</code>	The table to insert rows into.
<code>row_index</code>	The 0-based row index.
<code>column_index</code>	The 0-based column index.
<code>insert_right</code>	Whether to insert new columns to the right of the reference cell location. If True, cells will be inserted to the right of the cell reference. If False, cells will be inserted to the left of the cell reference.
<code>number</code>	The number of rows to be inserted. Maximum 20 per request.

---

`add_insert_table_rows_request`*Add an insert table row request*

---

**Description**

Add an insert table row request

**Usage**

```

add_insert_table_rows_request(
  google_slides_request = NULL,
  table_object_id,
  row_index,
  column_index,
  insert_below = TRUE,
  number
)

```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
table_object_id	The table to insert rows into.
row_index	The 0-based row index.
column_index	The 0-based column index.
insert_below	Whether to insert new rows below the reference cell location. If True, cells will be inserted below cell reference. If False, cells will be inserted above cell reference.
number	The number of rows to be inserted. Maximum 20 per request.

---

 add\_insert\_text\_request

*Add an insert text request*

---

**Description**

Add an insert text request

**Usage**

```

add_insert_text_request(
  google_slides_request = NULL,
  object_id,
  row_index = NULL,
  column_index = NULL,
  text,
  insertion_index = NULL
)

```

**Arguments**

<code>google_slides_request</code>	(Optional) A Google Slides Request object which is used to manage requests to the API
<code>object_id</code>	The ID of the object where the text is to be inserted into
<code>row_index</code>	(Optional) A numeric vector of row to insert the text into. This value is only optional for shapes
<code>column_index</code>	(Optional) A numeric vector of column to insert the text into. This value is only optional for shapes
<code>text</code>	A character vector of text which is to be inserted into the shape or table
<code>insertion_index</code>	(Optional) A numeric vector which indicate the starting point of how the text is to be inserted

---

`add_replace_all_shapes_with_image_request`  
*Add a replace all shapes with image request*

---

**Description**

Add a replace all shapes with image request

**Usage**

```
add_replace_all_shapes_with_image_request(  
  google_slides_request = NULL,  
  image_url,  
  replace_method = "CENTER_INSIDE",  
  page_object_ids = NULL,  
  text,  
  match_case = TRUE  
)
```

**Arguments**

<code>google_slides_request</code>	(Optional) A Google Slides Request object which is used to manage requests to the API
<code>image_url</code>	The image URL. The image is fetched once at insertion time and a copy is stored for display inside the presentation. Image must be less than 50mb in size
<code>replace_method</code>	The replace method. Accepts 'CENTER_INSIDE' and 'CENTER_CROP'
<code>page_object_ids</code>	(Optional) A character vector that contains the list of slide pages that you wish to enact the change on. It can contain multiple values.
<code>text</code>	The text to search for in the shape or table.
<code>match_case</code>	Indicates whether the search should respect case

**Examples**

```
## Not run:
library(rgoogleSlides)
rgoogleSlides::authorize()

# Define the presentation slide id (Can be retrieved from the url of the slides)
slides_id <- "<slide-id>"

# Define the internal drive image file to be inserted into slides
file_id <- "<file_id>"

request <- add_replace_all_shapes_with_image_request(image_url = file_id, text = 'aaa')
commit_to_slides(slides_id, request)

## End(Not run)
```

---

```
add_replace_all_text_request
      Add a replace all text request
```

---

**Description**

Add a replace all text request

**Usage**

```
add_replace_all_text_request(
  google_slides_request = NULL,
  replace_text = NULL,
  text = NULL,
  match_case = TRUE
)
```

**Arguments**

google_slides_request	(Optional) A Google Slides Request object which is used to manage requests to the API
replace_text	A character vector of text that would replace the ones in the text parameter. The order of the replaceText matters
text	A character vector of text that would be replaced by the ones in the replaceText parameter. The order of the text matters
match_case	(Optional) A boolean that takes in only TRUE or FALSE only. This would be applied across all requests

add\_update\_slides\_position\_request

*Add an update slides position request*

---

### **Description**

Add an update slides position request

### **Usage**

```
add_update_slides_position_request(  
    google_slides_request = NULL,  
    slide_object_ids = NULL,  
    insertion_index = NULL  
)
```

### **Arguments**

google\_slides\_request

(Optional) A Google Slides Request object which is used to manage requests to the API

slide\_object\_ids

A character vector of slide ids

insertion\_index

Numeric Vector. This is where the slides selected in slideobjectids parameter is inserted into. The slides would be inserted based on before the arrangement before the move.

---

aligned\_page\_element\_property

*Convenience function to return a page element property that aligns said element*

---

### **Description**

The Google Slides do not provide convenient ways to align the element. In order to do the necessary calculation, one has to be do take into account the slide size as well as image size.

### **Usage**

```
aligned_page_element_property(  
    slide_page_id,  
    slide_page_height = 5143500,  
    slide_page_width = 9144000,  
    image_height = NULL,  
    image_width = NULL,  
    align = "center"  
)
```

**Arguments**

<code>slide_page_id</code>	The id of the slide page that is to be altered
<code>slide_page_height</code>	The slide page height. It is set to default of 9144000
<code>slide_page_width</code>	The slide page width. It is set to default of 5143500
<code>image_height</code>	Image height in pt. Optional for align mode 'full'
<code>image_width</code>	Image width in pt. Optional for align mode 'full'
<code>align</code>	Alignment mode that is to be selected. 'center' or 'full' is accepted.

**Details**

The following pointers are thought of when doing the following calculations.

- The image is to be scaled without any distortions. No skews etc
- The translation coordinates of an image is defined by the top left corner of the image
- The following function will not query the Google Slides API to retrieve the page size automatically. The user would need to obtain that information and feed it into this function. This is to prevent this function from overutilizing the API unnecessarily when utilizing this function

---

authorize

*Authorize R package to access Google Slides API*

---

**Description**

This is a function to authorize the R package to access the Google Slides API. If no `client.id` and `client.secret` is provided, the package would provide predefined values.

**Usage**

```
authorize(
  client_id = getOption("slides.client.id"),
  client_secret = getOption("slides.client.secret"),
  token = NULL,
  ...
)
```

**Arguments**

<code>client_id</code>	OAuth client ID. This is obtained from Google API Credentials
<code>client_secret</code>	OAuth client secret. This is obtained from Google API Credentials
<code>token</code>	an output from <code>oauth2.0_token</code> to set as the authentication token.
<code>...</code>	additional arguments to send to <code>oauth2.0_token</code>

---

check_validity	<i>Convenience function to return a value if the value is NA</i>
----------------	--

---

**Description**

A function that checks and ensure that the value only returns null or a number. This function can only check one value at a time.

**Usage**

```
check_validity(value)
```

**Arguments**

value	Value to check if the value is valid. If value is NA, return as NULL instead.
-------	---

---

commit_to_slides	<i>Commit changes to slides</i>
------------------	---------------------------------

---

**Description**

Commit changes to slides

**Usage**

```
commit_to_slides(id, google_slide_request)
```

**Arguments**

id	The slide id of the googleslides to be updated
google_slide_request	The google slide request to be applied to the slides



---

create\_data\_table      *Create and upload data into a table in slides*

---

### Description

This function takes in one single page element property and one dataset at each time for uploading to a single slide

### Usage

```
create_data_table(  
  id = NULL,  
  page_element_property = NULL,  
  data = NULL,  
  headers = TRUE  
)
```

### Arguments

id	ID of the presentation slide
page_element_property	A list that contains a page element property. The page element is to be generated by the page_element_property function in this package.
data	Dataframe of the dataset that is to be converted so that it can be used with the google slides API
headers	Boolean to indicate whether to convert taking in mind of the headers or not

### Examples

```
## Not run:  
library(googleSlides)  
authorize()  
slide_page <- page_element_property(['slide-id-2'])  
create_data_table("presentation-slide-id", slide_page, iris[1:5,])  
  
## End(Not run)
```

---

create\_slides      *Create a new googleslide*

---

### Description

Create a new googleslide

**Usage**

```
create_slides(title = NULL, full_response = FALSE)
```

**Arguments**

title	Title of the presentation slide
full_response	Parameter to decide whether to return the full response or just the presentation ID

---

dataframe_convert	<i>Convert dataframe to dataframe with rows and columns</i>
-------------------	---

---

**Description**

Convert dataframe to dataframe with rows and columns

**Usage**

```
dataframe_convert(data = NULL, headers = TRUE)
```

**Arguments**

data	Dataframe of the dataset that is to be converted so that it can be used with the google slides API
headers	Boolean to indicate whether to convert taking in mind of the headers or not

---

get_endpoint	<i>Generate endpoint for the Google Slides API</i>
--------------	--

---

**Description**

Generate endpoint for the Google Slides API

**Usage**

```
get_endpoint(
  type_of_endpoint = "slides.endpoint.get",
  id = NULL,
  page_object_id = NULL
)
```

**Arguments**

type_of_endpoint	Type of endpoint to convert to url
id	(Optional) ID of the google slides to manipulate. Optional for slides.endpoint.create
page_object_id	(Optional) ID of the page slide object to be affected

---

`get_google_drive_urls` *Get the list of google drive url*

---

**Description**

Get the list of google drive url

**Usage**

`get_google_drive_urls(imageId)`

**Arguments**

`imageId` ID of the image on Google Drive

---

`get_slides_properties` *Get Google Slides Properties*

---

**Description**

Get Google Slides Properties

**Usage**

`get_slides_properties(id)`

**Arguments**

`id` ID of the presentation slide

---

`get_slide_page_properties`  
*Get a single page of a Google Slides property*

---

**Description**

Get a single page of a Google Slides property

**Usage**

`get_slide_page_properties(id, page_object_id, response = "simple")`

**Arguments**

id	ID of the presentation slide
page_object_id	The page ID of the presentation slide
response	Type of response. Values can be "simple" or "raw". A simple response provides a simplified object to query parts of the slide. A raw response provides the actual response via Google API.

---

is.google\_slide\_request

*Check if the object is a google slide request object*

---

**Description**

Check if the object is a google slide request object

**Usage**

is.google\_slide\_request(x)

**Arguments**

x	A google_slide_request object created from the rgoogleslides package
---	--

---

is.page\_element\_property

*Check if the object is a google slide request object*

---

**Description**

Check if the object is a google slide request object

**Usage**

is.page\_element\_property(x)

**Arguments**

x	A page_element_property object created from the rgoogleslides package
---	---

---

page\_element\_property *Common Property: Page Element Property*

---

### Description

This property is repeated in various of the request. Hence, to reduce the number of times this would appear in the code base, a function is created to handle the creation of this list.

### Usage

```
page_element_property(  
    page_object_id = NULL,  
    width_magnitude = NULL,  
    height_magnitude = NULL,  
    scale_x = NULL,  
    scale_y = NULL,  
    shear_x = NULL,  
    shear_y = NULL,  
    translate_x = NULL,  
    translate_y = NULL,  
    width_unit = "PT",  
    height_unit = "PT",  
    transform_unit = "PT"  
)
```

### Arguments

page_object_id	A character vector that contains the page id of the slide
width_magnitude	(Optional) A numeric vector that contains the width of the component
height_magnitude	(Optional) A numeric vector that contains the height of the component
scale_x	(Optional) A numeric vector that tells how the object is to be sized
scale_y	(Optional) A numeric vector that tells how the object is to be sized
shear_x	(Optional) A numeric vector that tells how the object is to be sheared
shear_y	(Optional) A numeric vector that tells how the object is to be sheared
translate_x	(Optional) A numeric vector that tells how the object is to be located
translate_y	(Optional) A numeric vector that tells how the object is to be located
width_unit	(Optional) A string vector that tells the measurement unit to use to locate the elements on the slide for the width element
height_unit	(Optional) A string vector that tells the measurement unit to use to locate the elements on the slide for the height element
transform_unit	(Optional) A string vector that tells the measurement unit to use to locate the elements on the slide for the transform element

**Value**

A PageElementProperty Object

**Examples**

```
## Not run:  
library(googleslides)  
  
# There is only one compulsory field which is pageObjectId is the  
# 'slide id' of the slide being referenced to  
pageElementProperty <- page_element_property("12345")  
  
## End(Not run)
```

---

rgoogleslides

*rgoogleslides: From R to Slides*

---

**Description**

Previously, when one is working with in the Google Ecosystem (Using Google Drive etc), there is hardly any good workflow of getting the values calculated from R and getting that into Google Slides. The normal and easy way out would be to just copy your work over but when you have a number of analysis to present with a lot of changes between each environment, it just becomes quite cumbersome.

Hence, this package is created to resolve such issues, providing a smoother workflow for such cases.

# Index

`add_create_image_request`, 2  
`add_create_shape_request`, 3  
`add_create_slide_page_request`, 4  
`add_create_table_request`, 5  
`add_create_video_request`, 6  
`add_delete_object_request`, 7  
`add_delete_table_column_request`, 8  
`add_delete_table_row_request`, 8  
`add_delete_text_request`, 9  
`add_insert_table_columns_request`, 10  
`add_insert_table_rows_request`, 10  
`add_insert_text_request`, 11  
`add_replace_all_shapes_with_image_request`,  
12  
`add_replace_all_text_request`, 13  
`add_update_slides_position_request`, 14  
`aligned_page_element_property`, 14  
`authorize`, 15

`check_validity`, 16  
`commit_to_slides`, 16  
`create_data_table`, 17  
`create_slides`, 17

`dataframe_convert`, 18

`get_endpoint`, 18  
`get_google_drive_urls`, 19  
`get_slide_page_properties`, 19  
`get_slides_properties`, 19

`is.google_slide_request`, 20  
`is.page_element_property`, 20

`oauth2.0_token`, 15

`page_element_property`, 21

`rgoogleSlides`, 22